ESANN 2011 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2011, i6doc.com publ., ISBN 978-2-87419-044-5.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100817300.

Training RBMs Based on the Signs of the CD Approximation of the Log-likelihood Derivatives

Asja Fischer[1] and Christian Igel[2] *

1- Institut für Neuroinformatik, Ruhr-Universität Bochum, Germany

2- Department of Computer Science, University of Copenhagen, Denmark

Abstract. Contrastive Divergence (CD) learning is frequently applied to Restricted Boltzmann Machines (RBMs), the building blocks of deep believe networks. It relies on biased approximations of the log-likelihood gradient. This bias can deteriorate the learning process. It was claimed that the signs of most components of the CD update are equal to the corresponding signs of the log-likelihood gradient. This suggests using optimization techniques only depending on the signs. Resilient backpropagation is such a method and we combine it with CD learning. However, it does not prevent divergence caused by the approximation bias.

1 Introduction

The rising field of deep learning led to a revival of Restricted Boltzmann Machines (RBMs) [1] as typical building blocks of Deep Belief Networks (DBNs, e.g., see[2]). Standard training of DBNs requires sequential training of RBMs in a layer wise fashion. Thus, effective and robust methods for RBM learning are a prerequisite for DBN training. Preforming maximum likelihood learning by vanilla steepest ascent is not possible in RBMs because the log-likelihood gradient involves averages which are exponential in the size of the smaller RBM layer and is thus not tractable. Therefore, *Contrastive Divergence* (CD, [3]) learning has become the standard learning algorithm.

The k-step CD update is based on steepest ascent on a biased estimate of the log-likelihood gradient gained by k steps of Gibbs sampling. The bias of the approximation depends on the mixing rate of the Markov chain, and mixing slows down with increasing absolute value of the model parameters [3, 4, 5, 6]. Hence, the bias increases with increasing parameter magnitude during RBM training. Recently, it has been shown that this can lead to divergence of the log-likelihood [7, 8, 9]. Nevertheless, by comparing the log-likelihood gradient and the expectation of the CD-k update in small RBMs (where both are tractable), Bengio and Delalleau [5] found that the fraction of parameter updates for which the log-likelihood derivatives and the corresponding CD-k components have different signs is small. Thus, optimization algorithms which consider only the signs of the partial derivatives could lead to better learning results.

*We acknowledge support from the German Federal Ministry of Education and Research within the National Network Computational Neuroscience, Bernstein Fokus: "Learning behavioral models: From human experiment to technical assistance", grant FKZ 01GQ0951.

ESANN 2011 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2011, i6doc.com publ., ISBN 978-2-87419-044-5.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100817300.

The speed of steepest ascent CD learning crucially depends on the learning rate (see for example the empirical results in [9]). For large-scale problems, optimization algorithms are needed that increase the likelihood in few iterations without extensive hyperparameter tuning. As the likelihood is in general intractable, the choice of the gradient-based optimization algorithm is limited to methods that do not need the absolute objective function value.

Resilient backpropagation (RProp, [10, 11]) is a powerful learning algorithm frequently used for neural network training. It autonomously adapts the step sizes for the parameter updates during learning. The algorithm depends only on the signs of the partial derivatives of the objective function and not on their absolute values. The standard variant does also not require the value of the objective function. Thus, RProp appears to be the ideal learning algorithm for training RBMs based on approximations of the log-likelihood gradient. It could suffer less from approximation errors, because it just requires the signs, and can adapt the learning rate automatically. Therefore, the combination of RProp and CD is empirically investigated in this paper. After briefly describing RBMs, CD-leaning, and RProp, we describe our experiments, discuss the results, and finally draw our conclusions.

2 RBMs and CD-Learning

An RBM is a bipartite undirected graphical model. The joint distribution of the m visible and n hidden (latent) variables under the model is $p(\boldsymbol{v}, \boldsymbol{h}) = e^{-\mathcal{E}(\boldsymbol{v},\boldsymbol{h})} / \sum_{\boldsymbol{v},\boldsymbol{h}} e^{-\mathcal{E}(\boldsymbol{v},\boldsymbol{h})}$. The energy $\mathcal{E}$ is given by $\mathcal{E}(\boldsymbol{v}, \boldsymbol{h}) = -\boldsymbol{h}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{v} - \boldsymbol{v}^{\mathrm{T}}\boldsymbol{b} - \boldsymbol{h}^{\mathrm{T}}\boldsymbol{c}$ with parameters $\boldsymbol{\theta} = (\boldsymbol{W}, \boldsymbol{b}, \boldsymbol{c})$, $\boldsymbol{W} \in \mathbb{R}^{n\times m}$, $\boldsymbol{b} \in \mathbb{R}^m$, $\boldsymbol{c} \in \mathbb{R}^n$.

The basic idea of the k-step Contrastive Divergence (CD-k) algorithm [3] is to run a Gibbs chain for only k steps starting from a training example $\boldsymbol{v}^{(0)}$ producing the sample $\boldsymbol{v}^{(k)}$. Each step t consists of sampling $\boldsymbol{h}^{(t)}$ from $p(\boldsymbol{h}|\boldsymbol{v}^{(t)})$ and sampling $\boldsymbol{v}^{(t+1)}$ from $p(\boldsymbol{v}|\boldsymbol{h}^{(t)})$ subsequently. The gradient with respect to $\boldsymbol{\theta}$ of the log-likelihood for $\boldsymbol{v}^{(0)}$ is then approximated by

$$\mathrm{CD}_k(\boldsymbol{\theta}, \boldsymbol{v}^{(0)}) = -\sum_{\boldsymbol{h}} p(\boldsymbol{h}|\boldsymbol{v}^{(0)}) \frac{\partial \mathcal{E}(\boldsymbol{v}^{(0)}, \boldsymbol{h})}{\partial \boldsymbol{\theta}} + \sum_{\boldsymbol{h}} p(\boldsymbol{h}|\boldsymbol{v}^{(k)}) \frac{\partial \mathcal{E}(\boldsymbol{v}^{(k)}, \boldsymbol{h})}{\partial \boldsymbol{\theta}} \quad . \qquad (1)$$

3 Training RBMs with Resilient Backpropagation

Resilient backpropagation is an iterative algorithm with adaptive individual step sizes [10]. It is frequently used for unconstrained optimization in machine learning because it is fast and robust with respect to the choice of the internal (hyper-) parameters, has linear time and space complexity in the number of parameters to be optimized, and is not very sensitive to numerical problems. The basic version of the RProp algorithm[1] considers only the signs of the partial derivatives of

[1] Usually we prefer the RProp variant called iRProp$^+$ [11]. However, this method requires the value of the objective function and not just the partial derivatives. Therefore, it is not well suited for the problem at hand.

ESANN 2011 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2011, i6doc.com publ., ISBN 978-2-87419-044-5.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100817300.

the function to be optimized and not their values. Because experiments suggest that the CD estimator has the correct sign most of the time [5], RProp seems to be promising for CD learning.

Let the ith component of the mean CD-approximation over the training set be denoted by $[\overline{\mathrm{CD}}_k(\boldsymbol{\theta}^{(g)})]_i$. In each iteration g of RProp, every parameter $\theta_i^{(g)}$ is updated according to

$$\theta_i^{(g+1)} = \theta_i^{(g)} + \operatorname{sign}\left([\overline{\mathrm{CD}}_k(\boldsymbol{\theta}^{(g)})]_i\right) \cdot \Delta_i^{(g)} \quad . \tag{2}$$

Prior to this update, the step size $\Delta_i^{(g)}$ is adapted based on changes of sign of the (approximated) partial derivative $[\overline{\mathrm{CD}}_k(\boldsymbol{\theta}^{(g)})]_i$ in consecutive iterations. If the sign changes, which indicates that a local minimum has been overstepped, then the step size is multiplicatively decreased, otherwise, it is increased. The update rule for the step size is given by:

$$\Delta_i^{(g)} = \begin{cases} \min(\eta^+ \Delta_i^{(g-1)}, \Delta_{\max}) & \text{if } [\overline{\mathrm{CD}}_k(\boldsymbol{\theta}^{(g-1)})]_i [\overline{\mathrm{CD}}_k(\boldsymbol{\theta}^{(g)})]_i > 0 \\ \max(\eta^- \Delta_i^{(g-1)}, \Delta_{\min}) & \text{if } [\overline{\mathrm{CD}}_k(\boldsymbol{\theta}^{(g-1)})]_i [\overline{\mathrm{CD}}_k(\boldsymbol{\theta}^{(g)})]_i < 0 \\ \Delta_i^{(g-1)} & \text{otherwise} \ , \end{cases} \tag{3}$$

where $0 < \eta^- < 1 < \eta^+$ and the step size is bounded by $\Delta_{\min}$ and $\Delta_{\max}$.

4 Experiments

We considered four artificial benchmark problems taken from the literature [12, 5, 9]. The *Labeled Shifter Ensemble* is a 19 dimensional data set containing 768 different samples and so the log-likelihood is $768 \log \frac{1}{768} \approx -5102.43$ if the distribution of the data set is modeled perfectly. We consider a variant of the *Bars and Stripes Ensemble* with 16 units and 32 input pattern yielding a bound of the log-likelihood of -102.59. Furthermore we considered the *Diag*$_d$-problem and the *1DBall*$_d$-problem with $d = 6$ dimensions. The first data set contains 7, the latter 24 unique binary vectors. The bounds for the log-likelihood are -13.62 and -76.27, respectively.

The RBMs were initialized with weights drawn uniformly from $[-0.5, 0.5]$ and zero biases. The numbers of hidden units were chosen to be equal to the number of visible units. The models were trained with RProp based on CD-1 or CD-100 on all four benchmark problems. If not stated otherwise, the hyperparameters where set to the default values $\eta^- = 0.5$, $\eta^+ = 1.1$, $\Delta_{\min} = 0.0$ and $\Delta_{\max} = 10^{100}$. To save computation time, the exact likelihood was calculated only every 10 iterations of the learning algorithm. All experiments were repeated 25 times.

5 Results

The left plot of figure 1 shows the evolution of the log-likelihood during learning of the Shifter-problem with RProp based on CD-1. After an increase in the first iterations the log-likelihood starts to decrease. The development of the

ESANN 2011 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2011, i6doc.com publ., ISBN 978-2-87419-044-5.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100817300.

likelihood differs a lot depending on the parameter initialization as can be seen exemplarily in the inset plot depicting some single trials. When using the CD-100 instead of the CD-1 approximation of the gradient a stagnation of the log-likelihood on an unsatisfying level during RProp based learning is observed (see right plot of Fig. 1). This happens systematically in every trail independent of the initialization of the parameters as indicated by the quartiles.

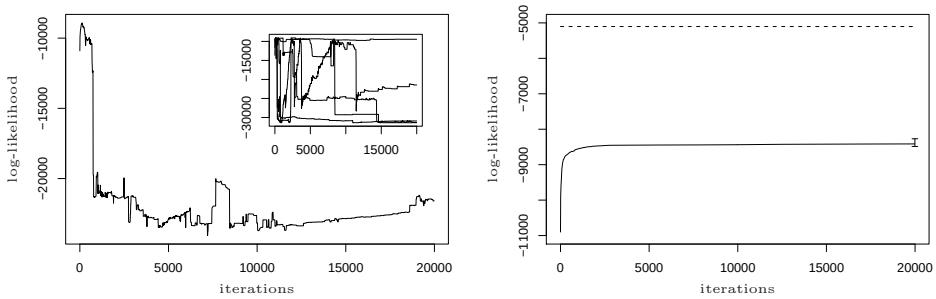

Fig. 1: The development of the log-likelihood during training RBMs on the Shifter-problem with RProp. Shown are the medians over 25 trails. Left: RProp based on CD-1. Five single trails with different parameter initializations are exemplarily shown in the inset plot. Right: RProp based on CD-100. Error bars indicate quartiles, the dashed line indicates the upper bound of the log-likelihood.

During training an RBM on the Bars-and-Stripes-problem the log-likelihood stagnates when RProp is based on CD-1 as well as on CD-100. In both settings similar log-likelihood values are reached which are low compared to the upper bound and to the maximum values reached when an RBM is trained with steepest ascent (see empirical analysis in [9]).

The log-likelihood also stagnates when learning the Diag- and 1DBall-problem. Here we also observe similar learning curves if the RProp algorithm is based on CD-1 and CD-100. The results are not shown due to the great similarity to the right plot of Fig. 1. For further empirical results we refer to the accompanying technical report [13].

The stagnation of the log-likelihood could indicate a frequently changing sign of the CD-approximation during the learning process. A frequently changing sign of the approximation causes the step size for the parameter updates (3) to get smaller and smaller and – if the step size is not bounded – to finally approach zero. Thus it could be possible to avoid the stagnation by enlarging the minimal possible step size $\Delta_{\min}$. This idea is verified by the following results.

If the minimal step size $\Delta_{\min}$ is set to a value larger than zero and the maximal step size $\Delta_{\max}$ is set to a value smaller than the default value, we can observe big differences in the evolution of the log-likelihood during RProp based training. As shown in Fig. 2, high (relative to the upper bound) log-likelihood values are reached during learning the Diag- and the 1DBall-problem

ESANN 2011 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2011, i6doc.com publ., ISBN 978-2-87419-044-5.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100817300.

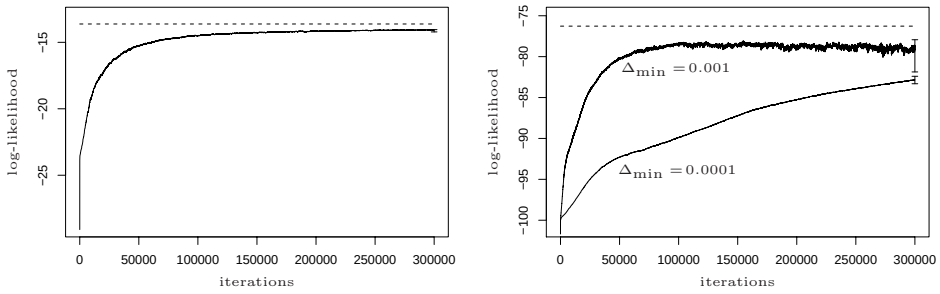

Fig. 2: Log-likelihood during training with RProp with limited step size hyperparameters. On the left: Learning of the Diag-problem. The step size is limited by $\Delta_{\min} = 0.0001$ and $\Delta_{\max} = 1$. On the right: Learning of the 1DBall-problem with $\Delta_{\max} = 1$ and $\Delta_{\min} = 0.001$ and $\Delta_{\min} = 0.0001$ respectively.

with RProp based on CD-1 if the hyperparameters are set to $\Delta_{\min} = 0.0001$ or $\Delta_{\min} = 0.001$, respectively, and $\Delta_{\max} = 1$. A nearly identical evolution of the log-likelihood can be observed (results not shown) if only the minimal step size value is enlarged and $\Delta_{\max}$ is set to its default value.

When learning the Bars-and-Stripes-problem with a restriction of the step size parameters ($\Delta_{\min} = 0.0001$ and $\Delta_{\max} = 1$) the log-likelihood starts to diverge. The experiments with the Shifter-problem with restricted step size parameters lead to similar results as the experiments with the parameters set to the default values.

6 Discussion and Conclusions

The experiments show that the success of RProp based CD learning depends on the data distribution to be learned and on the values of the hyperparameters ($\Delta_{\min}$ and $\Delta_{\max}$). If the step size is allowed to get arbitrary close to zero ($\Delta_{\min} = 0.0$), the training progress stagnated on an unsatisfying level for some target distribution. With an appropriate $\Delta_{\min}$, RProp was able to learn good models depending on the problem.

The reason for the stagnation could be convergence to a suboptimal local maximum. However, experiments using the expectation of CD-1 (not shown) did not suffer from the stagnation problem. As we see no reason why learning based on the expectation of CD-1 should be less prone to getting stuck in undesired local optima than learning based on the CD approximation, local maxima are not likely to be the reason.

We believe that the reason is the fast reduction of the RProp step size parameters Δ_i due to changes in sign of the gradient components induced by stochastic effects and errors in the CD approximation. Frequent changes of the sign could also be caused by ill-shaped log-likelihood functions [11]. However, if the RBMs were trained with RProp based on the exact likelihood gradient, good models for

ESANN 2011 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2011, i6doc.com publ., ISBN 978-2-87419-044-5. Available from http://www.i6doc.com/en/livre/?GCOI=28001100817300.

all benchmark problems could be learned in few iterations (results not shown).

If steepest ascent can learn a distribution reliably (e.g., when applied to Diag_6 or 1DBall_6), this is also possible using RProp, but in our experiments this required $\Delta_{\min} > 0$. In these cases, RProp may be preferable because $\Delta_{\min}$ may be easier to tune than learning rates in steepest ascent.

When applying RProp to Shifter and Bars-and-Stripes, the log-likelihood diverged before a good model was learned even if we constrained $\Delta_{\min}$ and $\Delta_{\max}$. That is, if learning diverges using steepest ascent (as reported in [9]), it also diverged using RProp. Thus, albeit it has been reported that the sign of the components of the CD update direction vector is often right, learning based on these signs tends to diverge.

In future work, we will evaluate RProp in combination with other approximations of the log-likelihood gradient (e.g., based on tempered transitions [8]).

References

[1] P. Smolensky. Information processing in dynamical systems: Foundations of harmony theory. In D. E. Rumelhart and J. L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1: Foundations*, pages 194–281. MIT Press, 1986.

[2] Y. Bengio. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 21(6):1601–1621, 2009.

[3] G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14:1771–1800, 2002.

[4] M. Á. Carreira-Perpiñán and G. E. Hinton. On contrastive divergence learning. In *10th International Workshop on Artificial Intelligence and Statistics (AISTATS 2005)*, pages 59–66, 2005.

[5] Y. Bengio and O. Delalleau. Justifying and generalizing contrastive divergence. *Neural Computation*, 21(6):1601–1621, 2009.

[6] A. Fischer and C. Igel. Bounding the bias of contrastive divergence learning. *Neural Computation*. In press.

[7] A Fischer and C. Igel. Contrastive divergence learning may diverge when training restricted Boltzmann machines. *Frontiers in Computational Neuroscience. Bernstein Conference on Computational Neuroscience (BCCN 2009)*, 2009.

[8] G. Desjardins, A. Courville, Y. Bengio, P. Vincent, and O. Dellaleau. Parallel tempering for training of restricted Boltzmann machines. *Journal of Machine Learning Research Workshop and Conference Proceedings*, 9(AISTATS 2010):145–152, 2010.

[9] A. Fischer and C. Igel. Empirical analysis of the divergence of Gibbs sampling based learning algorithms for Restricted Boltzmann Machines. In *International Conference on Artificial Neural Networks (ICANN 2010)*, LNCS. Springer-Verlag, 2010.

[10] M. Riedmiller. Advanced supervised learning in multi-layer perceptrons – From backpropagation to adaptive learning algorithms. *Computer Standards and Interfaces*, 16(5):265–278, 1994.

[11] C. Igel and M. Hüsken. Empirical evaluation of the improved Rprop learning algorithm. *Neurocomputing*, 50(C):105–123, 2003.

[12] D. J. C. MacKay. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, 2002.

[13] A. Fischer and C. Igel. Challenges in training Restricted Boltzmann Machines. In B. Hammer and T. Villmann, editors, *New Challenges in Neural Computation (NC^2)*, number 04/2010 in Machine Learning Reports, pages 11–24. 2010.